Correction TP d'introduction à Unix

Première année DUT Informatique

—2013-2014—

▸ Exercice 1. Connexion-Déconnexion

Ce premier exercice à pour but de vous familiariser avec votre environnement de travail. Vous pouvez passer les première étapes si vous êtes déjà connectés sur votre compte.

1. Allumez votre machine et choisissez de démarrer sous linux (Debian). Attention au démarrage si vous êtes dans la salle 220.

2. Lorsque que l'écran de connexion apparaît, entrez le *login* et le *mot de passe* qui vous ont été donnés avec votre carte d'étudiant. Si vous n'en n'avez pas encore, demander un compte temporaire à votre enseignant.

3. Une fois que vous êtes connectés, vous allez devoir lancer votre nouvelle application préférée: le **terminal**. Allez dans le menu `Applications`, puis dans le sous-menu `Accessoires` et vous y trouverez l'application `Termial`. Plutôt que de le lancer directement, faites un clic droit et choisissez: "ajouter ce lanceur au tableau de bord". Vous pouvez désormais directement lancer le terminal à partir de l'icône qui est apparue dans la barre en haut de l'écran.

4. Tapez la commande `pwd`. Ce qui s'affiche est le *nom absolu* de votre répertoire personnel. Cela vous indique le chemin allant de la racine du système de fichiers à ce répertoire. Comment pouvez vous revenir dans ce répertoire à partir de n'importe quel endroit dans le système de fichiers?

5. Créez un répertoire `Premier_test_repertoire` (notez que l'on n'utilise ni le caractère " ", ni les accents).

6. Quelle commande pouvez-vous utiliser pour vérifier que le répertoire a bien été créé?

7. Placez-vous dedans (grâce à la commande `cd`). Comme le nom est un peu long à taper, utilisez la touche `tab` pour le compléter.

8. Utilisez les flèches "haut" et "bas" pour faire défiler l'*historique* des commandes que vous avez déjà tapées et faites à nouveau afficher le nom absolu du répertoire où vous vous trouvez.

9. Sous Linux, la souris peut être utilisée pour faire les copier-coller, sans utiliser ni menu, ni clavier: double-cliquez sur le nom de répertoire que vous venez de faire afficher... puis cliquez avec le bouton du milieu. Le double clic sélectionne un mot (l'espace sert de séparateur) et le triple clic sélectionne toute la ligne.

10. À la fin de la séance, vous devrez **absolument** vous déconnecter (quitter votre compte). Vous êtes responsable de votre compte, si vous le laissez ouvert et que quelqu'un d'autre s'en sert (et efface tout ou envoie des mails en votre nom, par exemple), cela sera votre entière responsabilité. Pour vous déconnecter proprement (vous ne devez pas éteindre votre machine en appuyant sur le bouton de mise en route), allez dans le menu `Système` et choisissez "Fermer la session" ou "Eteindre" si vous êtes en fin de journée.

✂ -

ppoouueett

- ✂

▸ **Exercice 2. La commande man**

Construire un tableau suivant le modèle donné, contenant les commandes suivantes (avec ou sans option).

man, man -k,
cd, cd .., ls, ls -a, ls -l, ls -al, pwd,
mkdir, rmdir, cp, cp -R, mv, rm, rm -R,
touch, echo, history, history -c,
who, passwd,
cat.

Pour chacune de ces commandes, étudier la page d'aide et donner une description de la commande. Tester chacune de ces commandes sur des exemples concrets et analyser la réponse du système.

commande	Description Syntaxe Exemple
man	retourne le mode d'emploi de la commande s'il existe. `man [options] nom_commande` ex : `man man`
man -k	liste les commandes associées au mot clé donné. `man -k mot_cle` ex : `man -k copy`
cd	... ...

Si vous ne connaissez pas l'utilisation d'une commande, la recherche avec la commande man doit devenir un réflexe. Il est fortement conseillé de conserver ce tableau et de le compléter au fur et à mesure.

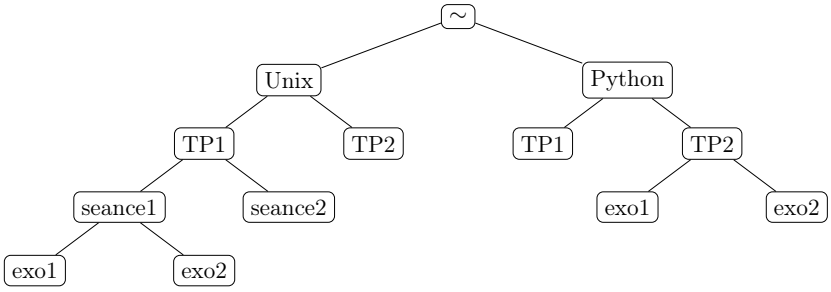
✂ -

man	retourne le mode d'emploi de la commande s'il existe. `man [options] nom_commande` ex : `man man`
man -k	liste les commandes associées au mot clé donné. `man -k mot_cle` ex : `man -k copy`
cd	permet de se déplacer dans le système de fichiers. `cd repertoire` ex : cd toto
cd ..	permet de se déplacer dans le répertoire parent.
ls	liste le contenu du répertoire courant. `ls [option] [fichier]` ex : `ls`
ls -a	liste le contenu du répertoire courant, sans ignorer les fichiers commençant par le symbole . `ls -a [fichier]` ex : `ls -a`
ls -l	liste le contenu du répertoire courant, en utilisant un format d'affichage long. `ls -l [fichier]` ex : `ls -l`
ls -al	liste le contenu du répertoire courant, sans ignorer les fichiers commençant par le symbole . et en utilisant un format d'affichage long. `ls -al [fichier]` ex : `ls -al`
pwd	affiche le nom du répertoire courant. `pwd [option]` ex : `pwd`
mkdir	crée un répertoire. `mkdir [option] repertoire` ex : `mkdir essai`
rmdir	supprime un répertoire vide. `rmdir [option] repertoire` ex : `rmdir essai`
cp	copie un fichier. `cp [option] source destination` ex : `cp essai TP1/`
cp -R	copie un répertoire de manière récursive. `cp -R source destination` ex : `cp -R TP1 TP2`
mv	déplace/renomme un fichier. `mv [option] source destination` ex : `mv essai TP1`
rm	efface un fichier. `rm [option] fichier` ex : `rm essai`
rm -R	efface récursivement un répertoire et son contenu. `rm -R fichier` ex : `rm -R TP1`

who	indique qui est connecté. `who [option] [fichier | arg1 arg2]` ex : `who`
echo	affiche une ligne de texte. `echo [option_courte] [chaine_de_caracteres]` ex : `echo "bonjour"`
passwd	modifie le mot de passe d'un utilisateur. `passwd [options] [LOGIN]`
history	montre à l'utilisateur les dernières commandes utilisées. `history nombre` ex : `history 5`
history -c	reinitialise l'history. `history -c` ex : `history -c`
touch	modifie les horaires des dernières modifications du fichier en les mettant à l'heure où la commande est appelée. Si le fichier passé en argument n'existe pas, le fichier est crée et est vide. `touch fichier` ex : `touch essai`
cat	concatène des fichiers et affiche le résultat sur le terminal. `cat [option] [file]` ex : `cat essai essai`

▸ Exercice 3. Manipulation de fichiers.

1. Créer dans votre répertoire personnel les fichiers selon l'arborescence suivante :



et de la manière suivante, chaque question doit être répondue en une seule commande et le répertoire courant doit toujours être votre répertoire personnel :

(a) Créer deux répertoires `Unix` et `Python` dans votre répertoire personnel.

(b) Créer deux répertoires `TP1` et `TP2` dans le répertoire `Unix`.

(c) Copier les deux répertoires `TP1` et `TP2` dans le répertoire `Python`.

(d) Créer deux répertoires `seance1` et `seance2` dans le répertoire `TP1` de `Unix`.

(e) Créer deux **fichiers** `exo1` et `exo2` dans le répertoire `seance1`.

(f) Copier les deux fichiers `exo1` et `exo2` dans le répertoire `TP2` de `Python`.

2. Placez vous dans le répertoire `Python/TP1`. Vérifiez que vous y êtes, puis, en une seule commande, placez vous dans le répertoire `Unix/TP2` en utilisant le *chemin relatif* du répertoire. Retournez dans `Python/TP1`, mais cette fois-ci, en utilisant le chemin absolu. Enfin, revenez à la racine de votre répertoire personnel.

3. Supprimer le répertoire `TP1` de `Python` avec la commande `rmdir`.

4. Supprimer le répertoire `TP2` de `Python` avec la commande `rmdir`, pourquoi la commande a échoué. Supprimer en une commande le répertoire `TP2` de `Python`.

5. Déplacer le fichier `exo1` dans le répertoire `TP1` et supprimer les répertoires `seance1` et `seance2`.

6. Renommer les fichiers `exo1` et `exo2` en respectivement `exercice1` et `exercice2`.

7. Dessiner l'arborescence des fichiers contenus dans le répertoire personnel de manière théorique. Vérifier le à l'aide d'une commande.

✂ ----------

1. (a) > `cd Python/TP1` > `pwd` > `cd ../../Unix/TP2` > `cd nom_du_chemin_absolu/Python/TP1` > `cd`

   (b) > `mkdir Unix Python`

   (c) > `mkdir Unix/TP1 Unix/TP2`

   (d) > `cp -R Unix/TP1 Unix/TP2 Python`

   (e) > `mkdir Unix/TP1/seance1 Unix/TP1/seance2`

   (f) > `touch Unix/TP1/seance1/exo1 Unix/TP1/seance1/exo2`

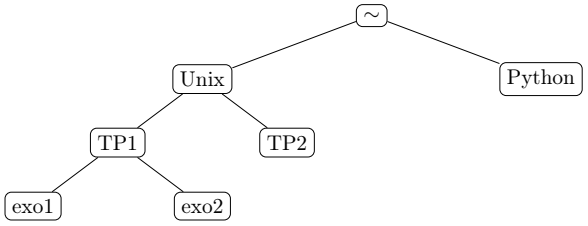
   (g) > `cp Unix/TP1/seance1/exo1 Unix/TP1/seance1/exo2 Python/TP2`

2. > `rmdir Python/TP1`

3. > `rmdir Python/TP2`
   La commande échoue car le dossier `TP2` n'est pas vide.
   > `rm -R TP2`

4. > `mv Unix/TP1/seance1/exo1 Unix/TP1/exo1`
   > `rm -R Unix/TP1/seance1 Unix/TP1/seance2`

5. > `mv Unix/TP1/exo1 Unix/TP1/exercice1`
   > `mv Unix/TP1/exo2 Unix/TP1/exercice2`



6. > `ls -R`

---------- ✂